UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/660,143	09/11/2003	Steven W. Githens	ROC920030276US1	4972

46797 7590 09/16/2009
IBM CORPORATION, INTELLECTUAL PROPERTY LAW
DEPT 917, BLDG. 006-1
3605 HIGHWAY 52 NORTH
ROCHESTER, MN 55901-7829

EXAMINER	
NUNEZ, JORDANY	

ART UNIT	PAPER NUMBER
2175	

MAIL DATE	DELIVERY MODE
09/16/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

**BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES**

Application Number: 10/660,143
Filing Date: September 11, 2003
Appellant(s): GITHENS ET AL.

____
Gero G. McClellan
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 05/18/2009 appealing from the Office action mailed 12/18/2008.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

20020156806	Cox	10-2002
6928436	Baudel	08-2005
7027056	Koselj	04-2006

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claims 1-4, 7-9, 11-17, 19-21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Cox et al. (US20020156806, Cox) in view of Baudel (US6928436).

As to claim 1, 2, 11, 19-21:

Cox shows a computer implemented method of generating a graphical representation of data, comprising:

receiving abstract attributes values comprising at least a selection of a requested graphical representation type (page 7, paragraph [0065],) for a selected data set (page 7, paragraph [0062]);

providing and selecting an abstract data structure template (e.g., Bar Chart view object) from a plurality of abstract data structure templates (e.g., dynamic tables, text objects) (abstract), each being specific to a different graphical representation type and defining a plurality of template attributes for generically representing an abstract graphical representation in the respective different graphical representation type, wherein the selected abstract data structure template is specific to the selected graphical representation type (page 7, paragraph [0072], lines 19-23);

generating, on the basis of the received abstract attributes values and the selected abstract data structure template, an abstract data structure defining a plurality of abstract attributes abstractly representing the data set in the graphical representation (figure 5, element 510, and corresponding text);

retrieving and providing transformation rules for transforming the abstract data structure into a concrete data structure, the transformation rules comprising a plurality of subsets of transformation rules each subset describing graphical attributes of a requested graphical representation type (page 5, paragraph [0044], lines 12-23);

selecting a subset of the plurality of subsets of transformation rules in accordance with a requested, graphical representation type (page 5, paragraph [0044], lines 12-23); an

selecting transformation rules (e.g., interaction desired programmed by author) for transforming the abstract data structure into a concrete data structure from a plurality of transformation rules, the transformation rules describing graphical attributes of the requested graphical representation type (page 5, paragraph [0044], lines 12-23);

and generating, on the basis of the abstract data structure and the selected subset of transformation, a concrete data structure defining a concrete graphical representation in a graphics rendering language using the transformation rules; wherein generating the concrete data structure is done by operation of a computer processor (figure 8, and corresponding description); and

transforming the abstract data structure into a plurality of concrete data structures, wherein transforming the abstract data structure is done by operation of a computer processor (page 8, paragraph [0080], lines 6-19).

Cox fails to specifically show: the transformation rules being specific to a different graphics rendering language, whereby the transformation rules support a plurality of graphics rendering languages; and transforming the abstract data structure into a plurality of concrete data structures, each concrete data structure corresponding to a different graphics rendering language.

In the same field of invention, Baudel teaches: a method for graphically rendering information of a database. Baudel further teaches: a visualization of information stored in a database (col. 1, l. 7-13), a visualization of a data table being a program that given as input any instance of a data table, outputs a uniquely defined sequence of graphic language instructions (col. 3, l. 48-50), a graphic language being a set of programming language functions and data types that enable describing images on a computer screen, examples of which OpenGL, Poscript, Java3D (col. 3, lines 22-29), and a DECORATION process setting graphic attributes for each record being drawn, said attributes including certain illumination models described in languages such as OpenGL and Java3D (i.e., because this process sets graphics attributes in languages such as OpenGL and Java3D, the visualization described inherently may be output in those languages).

Thus, it would have been obvious to one of ordinary skill in the art, having the teachings of Cox and Baudel at the time that the invention was made, to have combined the visualization of information stored in a database, a visualization of a data table being a program that given as input any instance of a data table, outputs a uniquely defined sequence of graphic language instructions, a graphic language being a set of programming language functions and data types that enable describing images on a computer screen, examples of which OpenGL, Poscript, Java3D, and a DECORATION process setting graphic attributes for each record being drawn, said attributes including certain illumination models described in languages such as OpenGL and Java3D of Baudel with the method as taught by Cox.

One would have been motivated to make such combination because a way to provide a simple but flexible user interface for accessing an available visualization would have been obtained and desired, as expressly taught by Baudel (column 1, lines 41-43).

As to claim 3, 13, Cox shows:

The method of claim 2, wherein the requested graphical representation type is one of a bar chart, a line chart, a pie chart, a scatter plot and a combination thereof (figure 8, and corresponding description).

As to claim 4, 14, Cox shows:

The method of claim 2, wherein the plurality of abstract data structure templates is associated with a particular data source of the data (page 7, paragraph [0059]).

As to claim 7, Cox shows:

The method of claim 1, wherein the requested graphical representation type is one of a bar chart, a line chart, a pie chart, a scatter plot and a combination thereof (page 8, paragraph [0079], lines 8-12).

As to claim 8, 16, Cox shows:

The method of claim 1, wherein at least one of the abstract data structure and the concrete data structure is defined in Extensible Markup Language (XML) (page 4, paragraph [0036], lines 4-7, page 10, paragraph [0102], lines 1-8).

As to Claims 9, 17, Baudel shows:

the concrete data structure is defined in a vector-based graphics language (c. 3, l. 26-29) (e.g., OpenGL).

As to claim 12, Cox shows:

The method of claim 11, further comprising:

rendering the data set, as described in the graphics rendering language (e.g., java applets), in a graphic (figure 8, and corresponding description).

As to claim 15, Cox shows:

The method of claim 11, wherein generating the concrete data structure using the transformation rules comprises:

selecting a subset of the transformation rules in accordance with the requested graphical representation type (page 5, paragraph [0044], lines 12-23);

and generating the concrete data structure using the subset of the transformation rules (page 8, paragraph [0080], lines 6-19).

Claims 10, 18, are rejected under 35 U.S.C. 103(a) as being unpatentable over Cox in view of Baudel, further in view of Koselj et al (US7027056, hereinafter Koselj).

As to Claims 10, 18:

Cox and Baudel show a method substantially as claimed, as specified above.

Cox and Baudel fail to specifically show: the vector-based graphics language is one of Vector Markup Language (VML), Scalable Vector Graphics (SVG), and Hypertext Markup Language (HTML) Image Maps.

In the same field of invention, Koselj teaches: a graphics engine and display driver integrated chip. Koselj further teaches: vector graphics language such as SVG, being used to send data to mobile and small area displays, while vector graphics language such as OpenGL being used for gaming APIs (i.e., SVG and OpenGL are obvious variations of vector graphic languages).

Thus, it would have been obvious to one of ordinary skill in the art, having the teachings of Cox, Baudel, and Koselj at the time that the invention was made, to have combined the vector graphics

language such as SVG being used to send data to mobile and small area displays, while vector graphics language such as OpenGL being used for gaming APIs of Koselj with the method as taught by Cox and Baudel.

One would have been motivated to make such combination because a way to have small-area displays which have size, weight, and power limitations properly display data would have been obtained and desired, as expressly taught by Koselj (column 1, lines 52-58).

References to specific columns, figures or lines should not be limiting in any way. The entire reference provides disclosure related to the claimed invention.

(10) Response to Argument

Appellant’s arguments have been fully considered but are not persuasive. Examiner reiterates that references to specific columns, figures or lines should not be limiting in any way. The entire reference provides disclosure related to the claimed invention.

Appellant argues:

a) Applicants respectfully submit that the Examiner's analogy fails to explain how the cited material, as well as Cox generally, discloses the above limitation. For example, the Examiner suggests that "raw data" in Cox teaches an abstract data structure. However, Applicants claim an "abstract data structure defining a plurality of abstract attributes representing a graphical representation of data." The Examiner is apparently relying on "raw data" in Cox to teach both data and abstract data structure defining a plurality of abstract attributes representing a graphical representation of data. That is, by treating the data and the abstract data structure defining a plurality of abstract attributes representing a graphical representation of data as being identical, the Examiner is wholly ignoring substantive limitations in the claims (namely, "defining a plurality of abstract attributes representing a graphical representation of data"), thereby fundamentally misconstruing the claims. On this basis alone, the Examiner's rejection is defective and should be withdrawn.

Examiner disagrees.

Appellant mischaracterizes Examiner’s analysis. For example, Cox (page 2, paragraph [0022]) clearly teaches "abstract data structure defining a plurality of abstract attributes representing a graphical representation of data" when, for example, it teaches an author sharing a database which is accessed by researchers, with the intention of creating visualizations for the data. As one of ordinary skill would readily understand, the data described must have a structure to it (e.g., a data structure) so that it can make sense both to the authors and researchers, and the structure inherently has attributes which are graphically represented when the data is visualized.

Appellant argues:

b) In addition, the Examiner fails to explain how the "raw data" in Cox "defines a plurality of abstract attributes representing a graphical representation of data." In fact, the "raw data" of Cox at best teaches data, and not abstract data structure. Further, Applicants claim "transformation rules for transforming the abstract data structure into a concrete data structure." For the reasons given above, Cox fails to teach "abstract data structure" as required by claim 1. Thus, it necessarily follows that Cox also fails to teach "transformation rules for transforming the abstract data structure." On this basis alone, the Examiner's rejection is defective and should be withdrawn.

Examiner disagrees.

Appellant’s conclusory statement (“Cox fails to teach [...]") may not be given weight in the absence of evidence. Also, see response to 1) above.

Appellant argues:

c) Further, even assuming, arguendo, that Cox somehow teaches "an abstract data structure defining a plurality of abstract attributes representing an abstract graphical representation of the data," the "actions" (i.e., a list of user command options) of Cox nevertheless fails to teach the recited subsets of transformation rules. See Advisory Action dated December 7, 2007 ("each action is a subset of a plurality of transformation rules (e.g., a view parameter may be changed, so that data displayed as a bar graph may be displayed as a pie chart) ...."). Respectfully, a user command option to "select specific data for

display" or to "change a view parameter" is not analogous to a transformation rule. Nevertheless, the Examiner states as follows: [The] Examiner believes it is rather clear that when a user issues a command to transform a bar graph into a pie chart, graphical attributes of a requested graphical representation type are indeed described, for example, a graphical attribute of a pie chart, which is a graphical representation, is described and then displayed to the user. Advisory Action dated February 20, 2009, page 2. In other words, the Examiner suggests that "a user command to display a pie chart" in Cox specifies graphical attributes of a requested graphical representation type (e.g., the pie chart). However, the user command to display a pie chart in Cox at best describes a requested graphical representation type (i.e., "to display a pie chart"), and not also graphical attributes of the requested representation type (e.g., graphical attributes of the pie chart). Further, the Examiner fails to explain how Cox teaches or suggests graphical attributes of the requested representation type. On this basis alone, the rejection is defective and should be withdrawn.

Examiner disagrees.

First, it is worth noting that Appellant does not take issue with the fact that Examiner analogizes a data visualization (e.g., a pie chart) with Appellant's “concrete data structure.” Second, Examiner still believes that it is rather clear that when a user issues a command to transform a bar graph into a pie chart, graphical attributes (e.g., visual characteristics) of a requested graphical representation type are indeed described, for example, a pie chart, would have graphical attributes, or visual characteristics, which are different from a pie chart.

Appellant argues:

d) Further, even assuming, arguendo, that the "actions" of Cox teaches graphical attributes of the requested representation type, Cox nevertheless fails to teach the subsets of transformation rules as claimed. In regards to the transformation rules, the Examiner states: [The] Examiner believes it is rather clear that in order to display a bar graph as a pie chart a subset of transformation rules take place which command the displayed bar to be transformed into the pie chart. Thus, list of a user command options or actions is indeed a subset of transformation rules when one of said user command options or actions is to

transform the bar graph into a pie chart. Advisory Action dated February 20, 2009, page 2. Even assuming, arguendo, that "bar graph" and "pie chart" teach the recited concrete data structure, Cox at best teaches "transforming a first concrete data structure into a second concrete data structure." Such is not the same as the recited transformation rules for transforming an abstract data structure into a concrete data structure. In other words, by treating the abstract data structure and the concrete data structure as being identical, the Examiner is wholly ignoring substantive limitations in the claims, thereby fundamentally misconstruing the claims. Therefore, Cox fails to disclose the recited transformation rules for transforming an abstract data structure into a concrete data structure. On this basis alone, the rejection is defective and should be withdrawn.

Examiner disagrees.

Appellant again mischaracterizes Examiner's analysis. As stated above in 1), it is data stored in a database that is visualized (e.g., made into bar graphs, pie chart), in other words, a visualization is not visualized. For example, when a user requests that a bar graph be displayed as a pie chart, one of ordinary skill in the art would understand that what actually happens is that the underlying data stored in the database is reprocessed so that a different visualization (e.g., a pie chart) is displayed instead of the original one (e.g., a bar graph). There is no "first concrete data structure" to "second concrete data structure" transformation.

Appellant argues:

e) Further, the Examiner suggests that Cox discloses the limitation of abstract data structure templates, each.., associated with a specific graphical representation type as recited in claim 2. Specifically, the Examiner asserts as follows: Cox shows a computer implemented method of generating a graphical representation of data, comprising.., providing and selecting an abstract data template (e.g., Bar Chart view object) from a plurality of abstract data structure templates (e.g., dynamic tables, text objects) (abstract) .... Final Office Action dated December 18, 2008, pages 2-3; see also Final Office Action dated September 25, 2007, page 3. That is, the Examiner suggests that the recited abstract data structure templates are disclosed by, e.g., a "BarChart view object." Id. Significantly, the Examiner's

analogy fails to conform to the other limitations of claim 2. That is, claim 2 requires a limitation of the abstract data structure being generated using the selected abstract data structure template. The Examiner analogizes the "raw data being analyzed as the abstract data structure." See Final Office Action dated September 25, 2007, page 7. Respectfully, the Examiner's analogy leads to a contradictory result and is therefore untenable. That is, the Examiner's analogy requires the raw data ("abstract data structure") to be generated using a view object such as "BarChart" ("abstract data structure templates"). Such a requirement is wholly contradictory and is not disclosed by (or even consistent with) Cox. That is, Cox does not describe "raw data" as being generated with the "view objects." Therefore, contrary to the Examiner's suggestion, Cox does not disclose abstract data structure templates, each.., associated with a specific graphical representation type. Nevertheless, the Examiner states: Cox teaches Bar chart[sic] view object. One of ordinary skill in the art would readily understand the Bar Chart View object to be used to generate a Bar chart. Further, one of ordinary skill in the art would readily understand that raw data is needed to generate a bar chart, and that the bar chart view object uses the raw data to generate display instructions to a display device, said display device ultimately displaying the bar chart. Thus, Cox clearly teaches "the abstract data structure["] (e.g., instructions to the display device) being generated using selected abstract[sic] data structure template (e.g., barch[sic] chart view object). Advisory Action dated February 20, 2009, page 2. However, the Examiner's rationale fails to confirm to the other limitations of claim 1. For example, the Examiner suggests that the "abstract data structure" includes "instructions to the display device." However, Applicants claim a concrete data structure (rather than the abstract data structure, from which the concrete data structure is generated) that "[defines] a concrete graphical representation of the data in a graphics rendering language." In other words, by again treating the abstract data structure and the concrete data structure as being identical, the Examiner is fundamentally misconstruing the claims. Thus, the Examiner's rationale is improper for suggesting that Cox teaches the abstract data structure being generated using the selected abstract data structure template as required by claim 2. Therefore, Cox fails to disclose the limitation of abstract data structure templates, each . . . associated with a specific graphical representation type as recited in claim 2. Accordingly, Applicants submit that the rejection is defective and should be withdrawn.

Examiner disagrees.

Appellant seemly does not take issue with the Examiner's statement that "one of ordinary skill in the art would readily understand that raw data is needed to generate a bar chart, and that the bar chart view object uses the raw data to generate display instructions to a display device, said display device ultimately displaying the bar chart." Instead, Appellant's argument proceeds by stating that Examiner suggests that an "abstract data structure" includes "instructions to the display device" while Appellant's invention recites a concrete data structure "[defines] a concrete graphical representation of the data in a graphics rendering language." As stated above, "the entire reference provides disclosure related to the claimed invention." It is not proper to argue that what an Examiner suggests is incorrect and then conclude that a prior art reference has been incorrectly applied. What is proper is to argue that the reference applied does not disclose the invention. In this case, the Examiner properly applies the prior art reference by stating the facts disclosed by Cox ("one of ordinary skill in the art would readily understand that raw data […]"). Further, the prior art of record discloses the invention, as evidenced by Appellant's lack of reference to what is actually disclosed by Cox, and instead relying on Appellant's interpretation of what "Examiner suggests."

Appellant argues:

f) Further, the Examiner suggests that Cox discloses generating, on the basis of the abstract data structure.., a concrete data structure defining a concrete graphical representation of the data in a graphics rendering language using the transformation rules as recited in claim 1. Specifically, the Examiner states: Cox shows a computer implemented method of generating a graphical representation of data, comprising . . generating, on the basis of the abstract data structure and the selected subset of transformation, a concrete data structure defining a concrete graphical representation in a graphics rendering language using the transformation rules; wherein generating the concrete data structure is done by operation of a computer processor (figure 8, and corresponding description) ... Final Office Action dated December 18, 2008, page 3; see also Final Office Action dated September 25, 2007, page 3. That is, the Examiner analogizes a "live document display output" of Cox, Figure 8 to teach generating.., a concrete data

structure .... In other words, the Examiner's interpretation merely equates a "concrete data structure" with an "display output." Respectfully, the Examiner's interpretation trivializes the limitation of "generating . . . a concrete data structure defining a concrete graphical representation of the data in a graphics rendering language" as recited in claim 1. Although a display output may be a result of rendering according to a graphics rendering language, a display output does not define anything in any graphics rendering language.

Examiner disagrees.

As acknowledged by Appellant, a display output may be a result of rendering according to a graphics rendering language. Thus, obviously, a disclosure of a display output (e.g., of a bar graph or a pie char, as disclosed by Cox) teaches a graphics rendering language.

Appellant argues:

g) Further, the description corresponding to Figure 8 in Cox (¶ 80-97) merely shows HTML code (i) for defining controls (¶ 83-87); and (ii) for invoking an applet (¶ 88-97). According to Cox: These controls are located within the text explaining their usage and in close proximity to the view bar chart view 832. Readers can also easily view the drivers with the most prize money by interacting with the "total winnings" histogram of view 834 or the related JavaScript controls .Cox, ¶ 80-81 and 94-97. As the cited passage illustrates, the controls in Cox are separate from the views in Cox, and are merely used to adjust the views. Therefore, following the Examiner's analogy (regarding the concrete data structure) yields a contradictory result, because the HTML code is not for generating a view (of the concrete data structure), but for defining a control, which is different from the view.

Examiner disagrees.

As acknowledged by Appellant in the first part of the argument, the HTML code is used for both defining a control and invoking an applet. Thus, unlike what Appellant states in the latter part of the argument, the HTML code is used not just for defining a control, but also for invoking an applet (which may contain a view of bar graph or a pie chart).

Appellant argues:

h) Further, the HTML code for invoking an applet merely invokes a Java applet (with two parameters: "url' and "Variable"). A Java applet is a software component that can run in a web browser. Moreover, the parameter "url value='drivers.txt'" is a reference to a text file containing raw data. In other words, the HTML code for invoking the applet is simply not any .graphical representation of data. That is, the HTML code for invoking the applet does not describe .graphics at all. Merely invoking a software component and supplying the software component with a filename of a file containing raw data is not the same as a concrete graphical representation in a graphics rendering language. Further, even assuming, arguendo, that Cox somehow discloses a concrete graphical representation in a graphics rendering language, Cox nevertheless fails to disclose that the concrete graphical representation is generated using the transformation rules, as required by claim 1. Therefore, for the reasons set forth above, individually and collectively, Cox does not disclose generating, on the basis of the abstract data structure.., a concrete data structure defining a concrete graphical representation of the data in a graphics rendering language using the transformation rules. Accordingly, Applicants respectfully submit that the rejection is defective and should be withdrawn.

Examiner disagrees.

Examiner agrees that “invoking a software component and supplying the software component with a filename of a file containing raw data is not the same as a concrete graphical representation in a graphics rendering language.” However, a reference must be viewed for all it teaches. Not only is the software component being invoked and supplied with data, but as a result of said invoking, Cox teaches (fig. 8, paragraphs [0094]-[0097]) a BarApplet (e.g., code=Idoc.BarApplet.Class) or Bar Graph is displayed, which obviously is produced a graphic rendering language

Appellant argues:

i) Even assuming, arguendo, that Baudel somehow teaches transforming the abstract data structure into a plurality of concrete data structures, Baudel nevertheless fails to disclose providing

transformation rules for transforming the abstract data structure into a concrete data structure, the transformation rules comprising a plurality of subsets of transformation rules each subset describing graphical attributes of a requested graphical representation type and being specific to a different graphics rendering language, as recited in claim 1. The Examiner suggests the contrary. Specifically, the Examiner states: Baudel explicitly teaches transformation rules (e.g., the rules used to transform a table of data into a visualization of said data), transforming the abstract data structure (e.g., table of data) into a concrete data structure (e.g., a visualization). Final Office Action dated December 18, 2008, page 8. Curiously, the Examiner does not provide any citation for the "explicit" teaching. In fact, Baudel in its entirety fails to disclose transformation rules of any sort. In other words, the Examiner is merely inferring a teaching of "transformation rules" from "creating a visualization of a table from a table of data using graphic language instructions" in Baudel. However, Applicants claim "transformation rules.., describing graphical attributes of a requested graphical representation type and being specific to a different graphics rendering language." Significantly, Baudel fails to disclose any transformation rules that describe graphical attributes of a requested graphical representation type or that are specific to a different graphics rendering language. Therefore, Baudel does not disclose providing transformation rules for transforming the abstract data structure into a concrete data structure, the transformation rules comprising a plurality of subsets of transformation rules each subset describing graphical attributes of a requested graphical representation type and being specific to a different graphics rendering language as recited in claim 1. Accordingly, Applicants respectfully submit that the rejection is defective and should be withdrawn.

Examiner disagrees.

As for the explicit teaching transformation rules for transforming the abstract data structure into a concrete data structure, Cox discloses this at least at fig. 8, paragraphs [0094] to [0097], where an abstract data “drivers.txt" which has been created using an abstract data template, is converted into a bar graph as show in figure 8. As far as the limitation, “transformation rules…describing graphical attributes of a requested graphical representation type,” Cox teaches (page 5, paragraph [0040] and [0041]) displaying the same abstract data in multiple formats (bar graph, pie chart), and one of ordinary skill in the art would readily understand that in order to display the same data in different formats there must be

“transformation rules…describing graphical attributes of a requested graphical representation type and being specific to a different graphics rendering language,” (e.g., there must be rules to turn said data into visualization, said rules comprising visual characteristics of the visualization, and said rules comprising instructions to a display for how to display said visualization). As far as the limitation “transformation rules… being specific to a different graphics rendering language” Baudel clearly teaches (col. 3, l. 22-29) this in its disclosure of different graphics rendering languages such as OpenGL, Poscript, and Java3D to visualize data.

Appellant argues:

j) The Examiner merely restates benefits of the references (e.g., "visualization of information"). With all due respect, it is not clear how the benefits are achieved or enhanced by virtue of the proposed modification. More fundamentally, the Examiner simply has not demonstrated how the references will be combined/modified. Instead, the Examiner simply posits that the references can be combined/modified on the basis of generally desirable features, such as visualization, without any specific explanation of how the references would synergistically interoperate to produce the claimed invention. Accordingly, the rejection is defective and should be withdrawn.

Examiner disagrees.

As to the synergism generated to produce the claimed invention, Baudel clearly teaches that it would have been desirable to make such modifications to Cox’s disclosure as a way to provide a simple but flexible (e.g., by virtue of using different graphics rendering languages) user interface for accessing an available visualization (col 1, l. 41-43).

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Jordany Nunez

/Jordany Núñez/

Examiner, Art Unit 2175

Conferees:

/William L. Bashore/

Supervisory Patent Examiner, Art Unit 2175

/DENNIS-DOON CHOW/

Supervisory Patent Examiner, Art Unit 2174